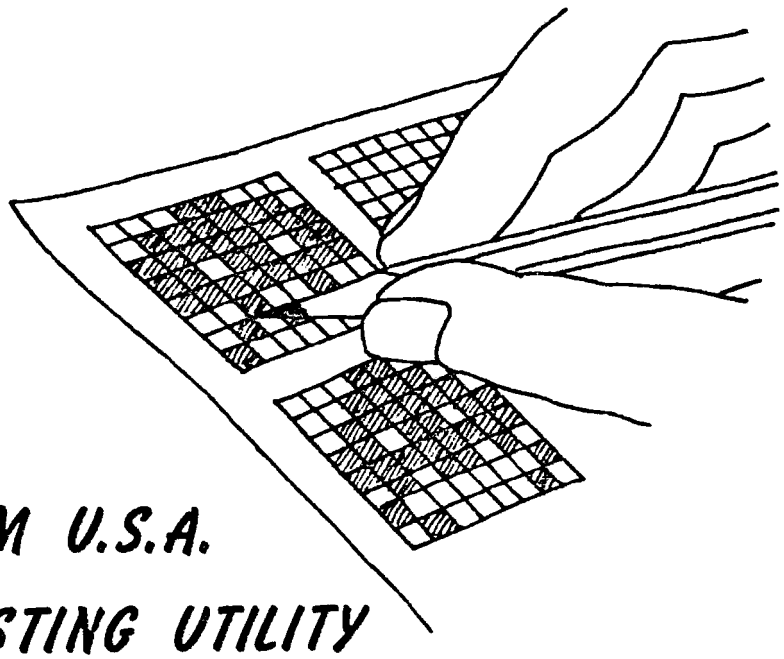
Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No. 5 February 1983

CHARACTER GRAPHICS



PLUS:

DOGGIES

REPORT FROM U.S.A.

PROGRAM LISTING UTILITY

HIDDEN SECRETS IN ATARI GAMES

AND MORE...

Atari Computer Enthusiasts (N.S.W.)

Atari Computer Enthusiasts (N.S.W.) is an independent, non-profit computer users' group loosely affiliated with Atari Computer Enthusiasts in the U.S.A. We have no connections with ATARI, Inc. or their Australian distributors, Futuretronics Australia Pty Ltd.

Our aims include promotion of the ATARI 400/800 Home Computer System; instructing both beginners and advanced users in programming techniques; exchanging hints, tips and ideas amongst members and generally enjoying ourselves.

Meetings are held at 6.0 P.M. sharp on the first Monday of every month (or the second Monday if it clashes with a public holiday) at the offices of:

I.P. Sharp Associates
8th Floor,
Carlton Centre,
55 Elizabeth Street,
Sydney,
(between King Street and Martin Place)

Membership to A.C.E.(N.S.W.) is $15 joining fee and $15 annual subscription (or $10 joining fee and $10 annual subscription for students under 18 and still at school).

Subscriptions or postal enquiries may be directed to:

Atari Computer Enthusiasts (N.S.W.)
G.P.O. Box 4514,
Sydney,
N.S.W. 2001

Phone enquiries:

Tony Reeve (President): 452-2974
Barry Williams (Vice President): 452-2229
Steven Marcus (Secretary/Treasurer): 387-4287
Garry Francis (Editor): 789-1397

Meeting Dates

7th March
*11th April (due to Easter)
2nd May
*6th June
4th July
*1st August
5th September
*10th October (due to 8-Hour Day Public Holiday)
7th November
*5th December

* indicates release dates for INSIDE INFO (subject to unforseen circumstances). Deadline for articles is the meeting prior to the release date.

QUESTION: If dogs reside in doggery and cats reside in cattery, then where do bugs reside?

ANSWER: Computer programs!

Editorial

The overriding theme for this issue is character graphics - a very powerful feature of the ATARI, but one which has been glossed over because of the ATARI's other more technically earth shattering abilities. Gregg Ramsey gets the ball rolling with an introduction to character graphics and supports this with 2 neat demos in GRAPHICS 0. We then present a practical application of character graphics in the form of a game from Stan Ockers of the U.S.A. It's in GRAPHICS 2, but don't let that fool you! This is the most professional game we've had so far - lots of colour and animation and a 7 page support article to discuss the game's design. The latter is essential reading to anybody wanting to write computer games.

Along a more serious vein, we've got an excellent utility for "paginating" program listings to the screen, a report on the ATARI scene in the U.S.A., details (at last) of the Software Exchange, lots of hints and tips and various other items of interest. Despite the promise in the last issue, Peter Stanhope has dropped his series on Copyright Law because he didn't think anybody was interested. Pity.

We were to have another game using character graphics in this issue, but in GRAPHICS 1. Unfortunately, it wasn't quite finished on time, so that'll be something to look forward to in the next issue. Also in the next issue, we are planning to continue the theme of character graphics by presenting a discussion of ANTIC modes 4 and 5. These are the multi-coloured character modes not directly supported by BASIC or the Operating System. We've got at least one game in ANTIC mode 4, so this should be an issue not to be missed.

Anyway, enough from me. See you all at the Computer Club Corner at the 1st Australian Personal Computer Show at Centrepoint, 10-12th March, 1983!

- Garry Francis

HIDDEN SECRETS IN ATARI GAMES

by Garry Francis

Missile Command! If you've got the Missile Command cartridge, you'll probably agree that it's a great game, but wouldn't it be just that little bit better if you could use a trackball like in the original arcade version to control the cursor? Well try this little test. Start the game the same as usual and watch how nice and smooth the cursor travels all around the screen. Now press CONTROL T. The screen should clear for a new game. Press START and try moving the cursor again. Holey mackeral! What happened? The cursor now only travels in little steps, but not in every direction. These characteristics lead me to believe that the program is now trying to read a trackball. There are a couple of trackballs available overseas, so if anyone gets one and my hunch proves to be correct, please let me know. Incidentally, return to joystick mode by pressing CONTROL T again. (Hmmm, I wonder if Centipede has a trackball mode?)

Caverns of Mars! Caverns of Mars was originally released through the ATARI Program Exchange quite some time ago. It didn't have the different difficulty levels (novice, pilot, etc.), but it had a useful feature which allowed you to skip to any of the levels you desired. If it wasn't for this feature, I'd never have seen past the space mines at Level 4. Unfortunately, this feature was omitted when ATARI took the program for its mainline range of software. Or was it? Try pressing the SHIFT, CONTROL and TAB keys simultaneously at any time after the game has started and you may get a pleasant surprise! (Ever wondered what Level 5 looked like?)

Super Breakout! This one was a real surprise! I gather that it is the programmer's dedication to his wife and son. Press the SHIFT, CONTROL and I keys (all at the same time) and up pops the message "I LOVE SUSIE AND BENJY TOO".

Has anyone else found any secret features in ATARI games? If so, let us all know about them.

INTRODUCTION TO CHARACTER GRAPHICS

by Gregg Ramsey

What is Character Graphics?

Character graphics is an easy way to create shapes to move around on the screen or to use for animation. GRAPHICS 0, 1 and 2 are often called "text" modes (although "character" modes is more correct) because, unlike the other GRAPHICS modes, they display characters instead of points. The letters, numbers and symbols that you type at the keyboard are all characters. When in GRAPHICS modes 0, 1 or 2, the computer knows to interpret the data in the display memory as ATASCII character codes rather than bit-mapped picture elements. However, you are not limited only to the letters and numbers you are already familiar with. This article explains how you can change the 'A' and have a space invader or pacman or whatever else you can think of appear on the screen in place of the 'A'. You can make mathematical symbols, superscripts or subscripts. You can have Russian or Chinese letters or design your own character fonts. For example, an Epyx game, "Temple of Apshai" uses character graphics to create a Medieval atmosphere. They have meticulously redefined every character and come up with an exellent Gothic character set which adds a professional touch to the whole game.

Unfortunately, character movement is limited to the size of the characters themselves which may make the movement quite "jerky" compared to the very smooth movement possible using other techniques (such as player-missile graphics). As these other techniques are sometimes quite difficult for the novice BASIC programmer, character graphics becomes quite an attractive alternative. As a matter of fact, character graphics can produce extremely high quality animation, i.e. where the shape of the object changes rather than just its position. For an example of this, take a look at ATARI's Space Invaders cartridge. Believe it or not, those active little invaders are just redefined characters in GRAPHICS 1!

The Character Set

Your faithful old ATARI prints an 'A' whenever you hit the 'A' key. Don't take this for granted though. There are a lot of processes carried out internally before that 'A' gets to the screen. For that matter, why should the 'A' shape get there at all? As I discussed in Inside Info No.2, a shape table of all the familiar ATARI characters resides in the Operating System ROM from location 57344 [$E000]. I won't go into an in depth discussion of this because it was fairly well covered before. Briefly though, a character is defined in an eight by eight grid with each row represented by one byte. Hence eight bytes are needed per character. A byte is eight BInary digiTs (BITs). When a bit is 0, the ATARI turns the corresponding pixel to the background colour. When it sees a 1, it "lights" the corresponding pixel. By examining all eight bytes for that character, it may be displayed on the screen.

As mentioned above, the character set is in ROM. This of course means that the characters can never be changed. Wouldn't it be neat though, if the ATARI could look somewhere else in memory for its characters. Glory Hallelujah! The designers of the ATARI have added just that feature which makes redefined character sets a simple task.

Custom Character Sets

When you hit the 'A' key, the ATARI says to itself, "He typed the 'A' key. Hmmm. I'll have to print a symbol to the screen, so I'll have to look up the character set to see what it looks like. But where's the character set? Ah huh! I'll just look at the Operating System pointer CHBAS located at 756 [$02F4] and see exactly where it starts." [Golly gosh! Is that what happens? - GF] CHBAS contains the page number of the start of the character set (where a page equals 256 bytes). The value in CHBAS would normally be 224, indicating that the character set starts at address 57344 (i.e. 256*224=57344). Since CHBAS is in RAM, POKEing different numbers into it will have it reading a different area of memory for its characters. For example, clear the screen and try POKE 756,0. It will think the character set starts at page zero and will fill the screen with gibberish, as this is how it now interprets the space character. Now type a quotation mark and you will see a character that is constantly changing. This is because the quote is described by bytes 16 to 23 of the

character set and these just happen to straddle RTCLOK at memory locations 18-20 [$12-$14]. This is ATARI's 3 byte real time clock and because it is being updated 50 times a second, the character will also change 50 times a second. Press SYSTEM RESET to return to normal.

To use custom characters, you must begin by getting some free memory not affected by the BASIC program, its various tables or the display memory. Since the area of free memory varies depending on the system configuration, you must set aside some RAM and protect it in such a way that it will work on any system. You must then copy the character set into your protected RAM using a FOR...NEXT loop, and change CHBAS to point to your shielded memory. Once this is done, the computer will do the same thing as before except that the character set is now in RAM and hence easily changed. It is then just a matter of finding the offset from the start of the character set for each individual character that you want to change and POKEing in the numbers to define its new shape.

The most common way to reserve RAM is to put it above the display list and display memory which are always at the top of user memory. An Operating System pointer called RAMTOP at location 106 [$6A] contains the page number of the first non-RAM byte in memory. Every time you do a GRAPHICS call, the ATARI looks into this location, clears out the area immediately below for the display memory and writes a new display list immediately below this.

We have now covered enough to present a simple BASIC routine for reading the character set into RAM:

```
10 A=PEEK(106):POKE 106,A-5
20 GRAPHICS 0:CUSTOM=256*(A-4)
30 FOR I=0 TO 1023
40 POKE CUSTOM+I,PEEK(57344+I)
50 NEXT I
60 POKE 756,CUSTOM/256
```

Line 10: Stores the value of RAMTOP into the variable A, then moves RAMTOP down by 5 pages. The new character set will only take up 4 pages of RAM, but we provide a 1 page buffer to allow for the clear screen command which would otherwise wipe out the first few characters. (See "Anatomy of Doggies" elsewhere in this issue for more details.)

Line 20: Do the GRAPHICS call (this may of course be GRAPHICS 1 or 2), then work out the address of the beginning of the new character set and store it in the variable CUSTOM. [Note that the character set must start on a 4 page boundary for GRAPHICS 0 or a 2 page boundary for GRAPHICS 1 and 2. In the case of GRAPHICS 1 and 2, it would only be necessary to use 2 pages for the new character set as these modes only utilise 64 characters rather than the 128 characters of GRAPHICS 0. - GF]

Lines 30-50: Do a FOR...NEXT loop to copy the old character set into our reserved area. When this is done, an exact copy of the character set in ROM will be in RAM.

Line 60: Start the computer reading from your new character set by taking the address of it, converting it into pages and POKEing it into the character set pointer, CHBAS.

You may now merrily change any character you like to whatever you can fit into an eight by eight grid. Refer to the examples for a method of doing this. I might just add that to get any real benefit from this article, you should type in the examples and analyse what you are typing as you do it. Maybe try to work out a better way of doing what I've done, but make sure you understand the code because there are animation techniques embedded in the programs.

This is definitely not the end of character graphics. There are 3 more modes that can't be accessed though BASIC's GRAPHICS command and they are the best. Anyway, until someone writes an article on these, see if you can put some of your own brilliant game ideas to good use armed only with ATARI BASIC and redefined characters.

```
1 REM ##############################
2 REM #      PACMAN'S REVENGE      #
3 REM #       by Gregg Ramsey      #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)    #
6 REM #        February 1983       #
7 REM ##############################
10 H=8:A=PEEK(106):POKE 106,A-5
20 GRAPHICS 0:SETCOLOR 2,0,0:POKE 559,
0:POKE 752,1:CUSTOM=256*(A-4)
30 POKE 16,64:POKE 53774,119
39 REM *** Redefine character set ***
40 FOR I=0 TO 1023:POKE CUSTOM+I,PEEK(
57344+I):NEXT I
50 POKE 756,CUSTOM/256
60 FOR I=97*H TO 106*H STEP H
70 FOR J=I TO I+7:READ D:POKE CUSTOM+J
,D:NEXT J:NEXT I:POKE 559,34
139 REM *** Ghost moving routine ***
140 FOR I=36 TO 4 STEP -4:A=I+20:B=I+3
0
150 FOR J=0 TO I:POSITION J,10:? " ab"
:POSITION J,11:? " cd"
160 FOR K=A TO B:SOUND 0,K,10,5:NEXT K
:FOR K=B TO A STEP -1:SOUND 0,K,10,5:N
EXT K:NEXT J:NEXT I
170 FOR I=1 TO 15:FOR J=A TO B:SOUND 0
,J,10,5:NEXT J:FOR J=B TO A STEP -1:SO
UND 0,J,10,5:NEXT J:NEXT I
180 FOR I=24 TO 0 STEP -2:A=I:B=I+10
190 FOR J=A TO B:SOUND 0,J,10,5:NEXT J
:FOR J=B TO A STEP -1:SOUND 0,J,10,5:N
EXT J:NEXT I
200 FOR I=0 TO 200:NEXT I
209 REM *** Pacman moving routine ***
210 FOR I=0 TO 35
220 POSITION I,10:? " ei":POSITION I,1
1:? " fj":FOR J=12 TO 25:SOUND 0,40-J,
12,10:SOUND 1,40-J,4,10:NEXT J
230 POSITION I+1,10:? " eg":POSITION I
+1,11:? " fh":FOR J=12 TO 25:SOUND 0,J
+15,12,10:SOUND 1,J+15,4,10:NEXT J
240 NEXT I:SOUND 0,0,0,0:SOUND 1,0,0,0
:FOR I=0 TO 320:NEXT I
250 POSITION 36,9:? CHR$(7):GOSUB 300:
POSITION 35,8:? CHR$(7):GOSUB 300:POSI
TION 30,7:? "BURP!"
260 FOR I=0 TO 100:SOUND 0,200,12,12:N
EXT I:POSITION 30,7:? "     ":POSITION
 35,8:? " ":POSITION 36,9:? " "
270 SOUND 0,0,0,0:POSITION 16,21:? "TH
E END":POSITION 6,23:? "PRESS SYSTEM R
ESET TO ABORT";
280 GOTO 280
300 FOR I=0 TO 15:NEXT I:RETURN
999 REM *** Data for characters ***
1000 DATA 3,12,16,36,46,93,81,74
1010 DATA 224,24,4,18,58,117,69,41
1020 DATA 68,80,76,67,64,70,73,112
1030 DATA 17,5,25,225,1,25,37,195
1040 DATA 7,31,60,125,127,255,255,255
1050 DATA 255,255,255,127,127,63,31,7
1060 DATA 224,248,252,254,254,255,255,
255
1070 DATA 255,255,255,254,254,252,248,
224
1080 DATA 224,240,224,192,192,128,128,
0
1090 DATA 0,128,128,192,192,224,240,22
4
```

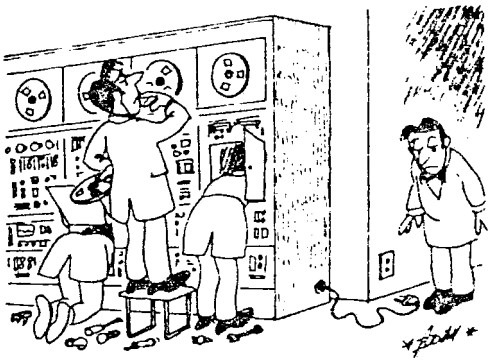
```
1 REM ##############################
2 REM #         CHESS BOARD          #
3 REM #       by Gregg Ramsey        #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #        February 1983         #
7 REM ##############################
10 DIM D$(1):H=8
20 A=PEEK(106):POKE 106,A-5
30 GRAPHICS 0:SETCOLOR 2,0,0:POKE 752,
1:CUSTOM=256*(A-4)
40 POKE 16,64:POKE 53774,119:REM Disab
le the BREAK key
50 POSITION 12,10:? "HANG ON A TICK."
60 POSITION 14,13:? "a b c d e f"
70 FOR I=0 TO 1023:POKE CUSTOM+I,PEEK(
57344+I):NEXT I:POKE 756,CUSTOM/256
150 FOR I=H*97 TO H*102 STEP 8
160 FOR J=I TO I+7:READ D:POKE CUSTOM+
J,D:NEXT J:NEXT I
170 FOR I=0 TO 500:NEXT I
200 GOSUB 2000
210 YPOS=4:GOSUB 500
220 YPOS=6:GOSUB 500
230 YPOS=16:GOSUB 500
240 YPOS=18:GOSUB 500
250 POSITION 6,23:? "PRESS SYSTEM RESE
T TO ABORT";
260 GOTO 260
500 FOR I=12 TO 26 STEP 2:READ D$:POSI
TION I,YPOS:? D$;:NEXT I:RETURN
1999 REM *** Draw board ***
2000 POKE 82,11:? CHR$(125):? :?
2010 ? CHR$(17);:FOR I=1 TO 7:? CHR$(1
8);CHR$(23);:NEXT I:? CHR$(18);CHR$(5)
2020 FOR I=1 TO 7
2030 FOR J=1 TO 9:? CHR$(124);CHR$(32)
;:NEXT J:?
2040 ? CHR$(1);:FOR J=1 TO 7:? CHR$(18
);CHR$(19);:NEXT J:? CHR$(18);CHR$(4)
2050 NEXT I
2060 FOR I=1 TO 9:? CHR$(124);CHR$(32)
;:NEXT I:?
2070 ? CHR$(26);:FOR I=1 TO 7:? CHR$(1
8);CHR$(24);:NEXT I:? CHR$(18);CHR$(3)
2080 RETURN
```

```
2498 REM xxx Data for characters xxx
2499 REM King
2500 DATA 24,126,24,90,255,126,60,255
2509 REM Queen
2510 DATA 24,189,255,126,60,60,126,255
2519 REM Bishop
2520 DATA 24,60,122,118,60,24,60,255
2529 REM Rook
2530 DATA 90,90,126,60,24,24,126,255
2539 REM Knight
2540 DATA 16,60,110,255,111,30,62,255
2549 REM Pawn
2550 DATA 0,0,0,24,60,24,60,126
2999 REM xxx Data for board set up xxx
3000 DATA d,e,c,a,b,c,e,d
3010 DATA f,f,f,f,f,f,f,f
3020 DATA f,f,f,f,f,f,f,f
3030 DATA d,e,c,a,b,c,e,f
```

Enhancements to SNAKE

In the rush and panic of getting SNAKE polished up for publication last issue, a couple of bugs crept in. This was partly my fault and I must offer my apologies to Gregg Ramsey. I'll make sure it never happens again by simply not printing a program until it's good and ready! Authors can do their bit by making sure that the program is indeed finished before submitting it. The following enhancements were suggested by Gregg:

- Any diagonal movement of the joystick causes the program to think that you are running into yourself and hence lose points. To correct this, change the IF statement in line 60 to read IF ST<7 OR ST=9 OR ST=10 OR ST=15 THEN GOTO NIL
- The volume is meant to drop whilst in the attract mode. To fix this, delete LOUD=8 from line 10, add LOUD=2 to line 420 and add LOUD=8 to line 3000.
- The rate given in the scoring window at the end of each game is out by a factor of 100. To correct this, delete /100 from the end of line 320.



DOGGIES

by Stan Ockers, Lockport, Illinois, U.S.A.

This one's for Johnny. At 1 1/2 years old he can't control the joystick yet, but he can push the button and he loves doggies. It's also hard enough to drive adults nuts. I know it can be done in 15 moves, but I haven't done it yet and I refuse to peek at the answer.

* * *

[This program first appeared in a different form in the excellent Eugene A.C.E. Newsletter, January 1982. Overseas subscriptions are $US20 per year. Send to A.C.E., 3662 Vine Maple Drive, Eugene, Oregon 97405, U.S.A. Instructions are included in the program from lines 2010 to 2060. It runs in 16k on a cassette based system and will also run in 16k on a disc based system if all the REMarks are deleted. Make sure you save the program before running it, as any typing errors in areas like the machine language subroutines could cause an unrecoverable system crash. Anything underlined should be typed in inverse video. For a full rundown on the program design and a discussion of some of the techniques used, see "Anatomy of Doggies" following the program listing. - GF]

```
1 REM ##############################
2 REM #          DOGGIES           #
3 REM #       by Stan Ockers       #
4 REM #  Eugene A.C.E. Newsletter  #
5 REM #        January 1982        #
6 REM #  Modified by Garry Francis #
7 REM # Reprinted by A.C.E.(N.S.W.) #
8 REM #        February 1983       #
9 REM ##############################
10 GOSUB 2000
19 REM xxx Main loop xxx
20 POKE 77,0:? #6;CHR$(125);"      dog
gies":? #6:? #6;" number of moves ";CH
R$(16)
30 P$="1110222":FOR C=1 TO 7:GOSUB 100
:IF A THEN GOSUB 210
40 NEXT C:POKE 209,120:POKE 53248,120:
MOVE=0:ATTRACT=0
50 IF STRIG(0)=0 THEN GOSUB 300:GOTO 5
0
60 IF PEEK(53279)=6 THEN POKE 209,0:PO
KE 53248,0:GOTO 20
70 C=INT(7*RND(0))+1:GOSUB 100:IF A TH
EN GOSUB 500+100*INT(3*RND(0))
80 GOTO 50+ATTRACT
99 REM xxx Which doggie? xxx
100 A=VAL(P$(C,C)):IF A=1 THEN LINE=10
00
110 IF A=2 THEN LINE=1100
120 X=3*C-3:RETURN
199 REM xxx Draw doggie xxx
200 FOR J=1 TO 14:NEXT J
210 RESTORE LINE:READ DOG$:POSITION X,
6:? #6;DOG$(1,2)
220 POSITION X,7:? #6;DOG$(3,4):POSITI
ON X,8:? #6;DOG$(5,6):RETURN
299 REM xxx Process player's move xxx
300 POKE 77,0:B=0:C=INT(PEEK(209)/24-0
.5):IF C<1 THEN C=1
310 IF C>7 THEN C=7
320 B=B+1:IF VAL(P$(B,B)) THEN 320
330 IF B=C THEN RETURN
340 IF C<B-2 OR C>B+2 THEN GOSUB 100:G
OSUB 800:RETURN
350 TEMP=C:FOR C=1 TO 7:IF C=TEMP THEN
 400
360 GOSUB 100:IF A=0 THEN 400
370 IF C<TEMP THEN LINE=LINE+6
380 IF C>TEMP THEN LINE=LINE+7
390 GOSUB 210
400 NEXT C:C=TEMP:GOSUB 100:GOSUB 500:
GOSUB 600:LINE=LINE+8:V=8:INC=0:GOSUB
900:LINE=LINE+2:INC=-1:GOSUB 900
410 LINE=LINE+2:GOSUB 900:POSITION X,6
:? #6;"  ":P$(B,B)=P$(C,C):P$(C,C)="0"
:C=B
420 GOSUB 100:LINE=LINE+12:V=1:INC=1:G
OSUB 900:LINE=LINE-2:GOSUB 900:LINE=LI
NE-10:GOSUB 700:GOSUB 600
430 MOVE=MOVE+1:MOVE$=STR$(MOVE):POSIT
ION 17,2:FOR I=1 TO LEN(MOVE$):? #6;CH
R$(ASC(MOVE$(I,I))-32);:NEXT I
440 IF P$<>F$ THEN RETURN
450 POSITION 2,4:IF MOVE=15 THEN ? #6;
"  great stuff":GOTO 480
460 IF MOVE<20 THEN ? #6;"   good goin
g":GOTO 480
470 ? #6;"could be better"
480 POSITION 0,11:? #6;"press start to
 begin";:ATTRACT=10:RETURN
499 REM xxx Bark xxx
500 LINE=LINE+5:GOSUB 210:FOR I=1 TO 6
:SOUND 0,ASC(BARK$(I)),12,14-I*2:SOUND
 1,ASC(BARK$(I)),14,I*2:NEXT I
510 LINE=LINE-5:GOSUB 210:SOUND 0,0,0,
0:SOUND 1,0,0,0:RETURN
599 REM xxx Wag tail xxx
600 LINE=LINE+1:FOR I=1 TO 3:LINE=LINE
+1:GOSUB 200:LINE=LINE-1:GOSUB 200:NEX
T I:LINE=LINE-1:GOSUB 200:RETURN
699 REM xxx Hop xxx
700 LINE=LINE+3:V=8:INC=0:GOSUB 900:LI
NE=LINE-3:GOSUB 200:RETURN
799 REM xxx Shake head xxx
800 FOR I=1 TO 3:LINE=LINE+6:GOSUB 200
:LINE=LINE-6:GOSUB 200
810 LINE=LINE+7:GOSUB 200:LINE=LINE-7:
GOSUB 200:NEXT I:RETURN
899 REM xxx Footsteps sound xxx
900 FOR I=1 TO 3:V=V+INC:LINE=LINE+1:G
OSUB 200:SOUND 0,6,13,V:SOUND 0,0,0,0
910 LINE=LINE-1:GOSUB 200:SOUND 0,11,1
3,V:SOUND 0,0,0,0:NEXT I:RETURN
999 REM xxx Shapes of doggies xxx
1000 DATA '()x%&
1001 DATA '()\%&
1002 DATA '([x%&
1003 DATA '([x%-
1004 DATA '()\+&
1005 DATA !"#$%&
1006 DATA ./:;%&
1007 DATA WY<=%&
1008 DATA >?Z^+&
1009 DATA >?]_%-
1010 DATA @HJK__
1011 DATA @HJK__
1012 DATA XQ____
1013 DATA XQ____
1100 DATA '()x%&
1101 DATA '()\%&
1102 DATA '([x%&
1103 DATA '([x%-
1104 DATA '()\+&
1105 DATA !"#$%&
1106 DATA ./:;%&
1107 DATA WY<=%&
1108 DATA >?Z^+&
1109 DATA >?]_%-
1110 DATA @HJK__
1111 DATA @HJK__
```

```
1112 DATA XQ___
1113 DATA XQ___
1999 REM xxx Initialisation xxx
2000 GRAPHICS 21:START=PEEK(106)-4:POK
E 106,START-1:GRAPHICS 0:POKE 710,0:PO
KE 709,12:POKE 752,1:POKE 82,1
2009 REM xxx Instructions xxx
2010 POSITION 16,1:? "DOGGIES":? :? "T
here are 3 white doggies on the left"
2020 ? "side of the screen and 3 brown
 doggies":? "on the right side.  You m
ust reverse"
2030 ? "their position by moving one d
oggie at":? "a time.":? :? "Use the jo
ystick in Port 1 to put the"
2040 ? "bone under the doggie you wish
 to":? "move, then press the fire butt
on.  The"
2050 ? "doggie will move into the empt
y space,":? "but only if he is next to
 it or no"
2060 ? "more than one doggie away.":?
:? "It can be done in 15 moves!  Can y
ou":? "do it?"
2070 POSITION 13,22:? "_INITIALISING_"
:DIM BARK$(6),ML$(32),DOG$(6),MOVE$(3)
,F$(7),P$(7):F$="2220111"
2079 REM xxx Data for barking xxx
2080 RESTORE 2090:FOR I=1 TO 6:READ A:
BARK$(I)=CHR$(A):NEXT I
2090 DATA 97,109,97,9,5,5
2099 REM xxx VBI service routine xxx
2100 FOR I=1536 TO 1578:READ A:POKE I,
A:NEXT I
2110 DATA 104,160,10,162,6,169,7,76,92
,228,173,0,211,72,41,8,208,6,230,209,1
04,24,144,7,104,41,4,208,11,198,209
2120 DATA 169,0,133,77,165,209,141,0,2
08,76,98,228
2129 REM xxx P-M Graphics xxx
2130 POKE 54279,START:POKE 53256,1:PM=
256*START:POKE 53248,0:POKE 209,0:POKE
 704,30
2140 FOR I=PM+602 TO PM+604:READ A:POK
E I,A:NEXT I
2150 DATA 195,255,195
2159 REM xxx Move character set xxx
2160 FOR I=1 TO 32:READ A:ML$(I)=CHR$(
A):NEXT I:CH=256*START:X=USR(ADR(ML$),
57344,CH)
2170 DATA 104,104,133,204,104,133,203,
104,133,206,104,133,205,162,2
2180 DATA 160,0,177,203,145,205,136,20
8,249,230,204,230,206,202,208,240,96
2189 REM xxx Redefine characters xxx
2190 READ X:IF X=-1 THEN 2560
2200 FOR I=CH+X TO CH+X+7:READ A:POKE
I,A:NEXT I:GOTO 2190
2210 DATA 8,7,15,31,61,109,111,110,111
2220 DATA 16,224,240,248,188,182,246,1
18,246
2230 DATA 24,12,12,7,3,3,7,15,31
2240 DATA 32,48,48,224,192,192,224,240
,248
2250 DATA 40,31,31,31,31,31,24,248,248
2260 DATA 48,248,248,248,248,248,24,31
,31
2270 DATA 56,0,7,31,63,125,237,239,206
2280 DATA 64,0,224,248,252,190,183,247
,115
2290 DATA 72,15,12,15,7,3,7,15,31
2300 DATA 80,240,48,240,224,192,224,24
0,248
2310 DATA 88,31,31,31,31,27,248,248,0
2320 DATA 104,248,248,248,248,216,31,3
1,0
2330 DATA 112,0,7,15,31,63,63,55,55
2340 DATA 120,0,128,192,192,64,127,255
,255
2350 DATA 208,55,7,7,3,3,7,15,31
2360 DATA 216,254,224,254,252,192,224,
240,248
2370 DATA 224,127,7,127,63,3,7,15,31
2380 DATA 232,236,224,224,192,192,224,
240,248
2390 DATA 240,0,7,31,63,127,239,239,20
7
2400 DATA 248,0,224,248,252,254,247,24
7,243
2410 DATA 256,1,7,15,11,3,1,3,7
2420 DATA 320,128,224,240,208,192,128,
192,224
2430 DATA 336,7,7,7,7,7,4,28,0
2440 DATA 344,224,224,224,224,224,32,5
6,0
2450 DATA 392,192,192,128,192,192,192,
192,96
2460 DATA 440,0,1,3,3,2,254,255,255
2470 DATA 448,3,3,1,3,3,3,3,6
2480 DATA 456,0,224,240,248,252,252,23
6,236
2490 DATA 464,15,15,15,7,3,7,15,31
2500 DATA 472,15,12,15,71,67,71,47,31
2510 DATA 480,240,48,240,226,194,226,2
44,248
2520 DATA 488,15,15,15,71,67,71,47,31
2530 DATA 496,240,240,240,226,194,226,
244,248
2540 DATA 504,240,240,240,224,192,224,
240,248
2550 DATA -1
2559 REM xxx Return to user xxx
2560 POSITION 9,22:? "Press _START_ to
 begin":POKE 53279,8
2570 FOR I=1 TO 20:IF PEEK(53279)<>6 T
HEN NEXT I:POKE 755,2-PEEK(755):GOTO 2
570
2580 POP :GRAPHICS 18:POKE 16,64:POKE
53774,119:POKE 756,START:POKE 559,46:P
OKE 53277,3
2590 POKE 708,14:POKE 709,184:POKE 710
,20:POKE 711,136:A=USR(1536):RETURN
```

ANATOMY OF DOGGIES

by Garry Francis

Who The Hell Is Stan Ockers Anyway?

Way back in June 1981, the Eugene A.C.E. Newsletter published a lunar lander program written by a fellow named Stan Ockers. This in itself was not unusual, but Stan returned with another program in the next issue, and the next, and the next, and the next... In fact, this prolific programmer from Lockport, Illinois has now written a program (or sometimes two) for every single issue of the Eugene A.C.E. Newsletter since that issue in June 1981.

In December 1981, editor Mike Dunn awarded Stan with the First Editor's Award. His work is often reprinted in ANTIC and the Michigan A.C.E. Newsletter and some programs (such as Chicken) are already considered classics.

Stan's programs may be divided into 2 very distinct categories - games and utilities. They are like a developing series of tutorials in advanced graphics techniques from BASIC. His philosophy has been that "...perhaps the best way to explain what is going on is to go through an actual program...". But even if you don't want to learn about fancy graphics, the programs are still fun to use.

I have always admired Stan Ockers' work, but I was astounded to find out that nearly all his programs were developed on a cassette based ATARI 800 with 16k RAM and a black and white TV! He didn't even have an assembler! The moral is, of course, that you don't need 256k and a quadruple density disc drive to write good programs, as many people seem to think. You merely need a good imagination and a lot of hard work.

Enter Doggies

Doggies has always been one of my favourite of Stan Ockers' games. I originally looked through the code to see how on earth it worked. Just recently, I decided that it would be an excellent program for the magazine. Stan has often pointed out that some of his programs have minor deficiencies (such as lack of colour), which he leaves for readers to fix themselves. So I set about to do a bit of polishing up to Doggies. Unfortunately, my polishing up got out of hand and I have now completely restructured the program, removed a couple of minor bugs, improved the response time, screen display and colours and renumbered it. But despite the weeks of work I put into it, it is still Stan Ockers' program. He should be congratulated on doing such a fine job and making my task such a pleasure.

Program Flow

Doggies begins by jumping to a massive block of initialisation code which is placed at the end of the program so as not to slow things down when the main program is executing later on.

I believe that the first thing you should do in any program is to have something happen immediately after typing RUN, even if you only clear the screen and print the title. A pause of 2 or 3 or even 20 seconds before anything happens is totally unacceptable, as the user may think the program has crashed. You should also make no assumptions as to screen conditions before the program was run. Therefore, as a rule of thumb, always start your initialisation with a GRAPHICS statement, followed by the necessary POKEs and SETCOLOR statements to set margins, colours and so on, even if you are using default values. Doggies also clears out and reserves an area of memory at the top of RAM (see Memory Allocation), then prints the instructions and the word "INITIALISING" so that you know what's going on. You can now read the instructions oblivious of the fact that the remainder of the initialisation is being carried out. This takes about 11 seconds. When finished, "INITIALISING" is overprinted by another message to "Press START to begin". When you've done so, the screen will be cleared, player-missile graphics are enabled and the vertical blank interrupt routine is set up. The BREAK key is also disabled to force you to use SYSTEM RESET to abort the program. This is the only way to ensure that all system parameters are back to normal when you return to BASIC. Whew!

We now set up the screen for a new game and cycle through the main program loop from lines 50 to 80. Yes, that's right! The main program loop is a mere 4 lines long! The remainder of the program is subroutines for specific actions. Each subroutine commences on a line number of a multiple of 100 and is preceded by a REM to indicate its function. I will delve into some of these later. The remainder are fairly straightforward.

Doggies uses a number of the ATARI's special features such as interrupt processing, player-missile graphics and a redefined character set. We'll now take a look at some of these aspects, but be warned that a good working understanding of BASIC is assumed.

Memory Allocation

As a lot of our members are beginners with minimum systems, I intended right from the start that Doggies should run on a cassette based system with only 16k of memory. It does this quite admirably, but if you've got a disc based system, you'll need to delete all the REMarks if you expect to run it in 16k. (You shouldn't really be using a disc drive with only 16k anyhow, as there is just too little room left for programs or high resolution GRAPHICS modes.)

Figure 1 shows the maximum memory requirements for Doggies when typed in exactly as shown in the listing. RAMTOP is an Operating System pointer at memory location 106 [$6A] which tells us the page number of the first non-RAM byte in memory. (A page is a block of 256 bytes.) By POKEing a lower value into RAMTOP, we can fool the system into thinking that it has less RAM than it actually has. When we carry out a GRAPHICS command, the display memory and display list will be written below the new value of RAMTOP, thereby reserving an area for our own use. We use this technique in line 2000 to set aside an area for player-missile graphics and the redefined character set.

The player-missile graphics require 4 pages (or 1024 bytes) of memory for double line resolution and must start on a 1k boundary. We define the starting location with the variable START. The first 384 bytes of this area (i.e. START to START+383) are not used. The area from START+384 to START+511 is reserved for missiles, but we will not be using them. It seems a shame to waste 512 bytes, so seeing that our redefined character set also requires 512 bytes, we can store it in this area. We also need a 1 page buffer between the display memory and START due to the RAMTOP dragon who gobbles up the first 64 bytes above RAMTOP whenever a GRAPHICS command or clear screen command are executed.

The GRAPHICS 0 display used for the instructions is shown in Figure 1 as it is more memory intensive than the GRAPHICS 18 display used for the game itself. Just out of interest, GRAPHICS 18 only requires 20 bytes for its display list and 240 bytes for its display memory.

You can see from Figure 1 that the BASIC program itself requires 7245 bytes just to reside in memory. (The structure of the tokenised BASIC program is not important at this point.) When run, it will then take up an extra 61 bytes for string storage plus a variable amount for the run-time stack (this keeps track of return addresses for GOSUB's and FOR...NEXT loops). The area labelled "DOS" is only applicable if you've got a disc drive booted. The figures are for DOS 2. The area labelled "Operating System RAM" is reserved by the Operating System for a variety of functions. (Maybe we could delve into all this stuff in a future issue.) By adding up the number of bytes used by your system, you can see that a cassette based system requires well under 16k (note that 16k is 16384 bytes), whereas a disc based system requires just over 16k, hence the need to delete the REMarks for a disc based system with only 16k.

Player-Missile Graphics

Whenever using player-missile graphics, you must first reserve an area (as discussed above) to store the images and then clear this area of any extraneous garbage. I used a little trick to accomplish the latter by issuing a GRAPHICS 21 command prior to reserving the P-M graphics area (see line 2000). This has the effect of clearing out 960 bytes at the top of memory for the GRAPHICS 21 display. When RAMTOP is moved down, the 640 bytes used by P-M graphics remains undisturbed and set to all zeroes. GRAPHICS 0 or 5 could also be used

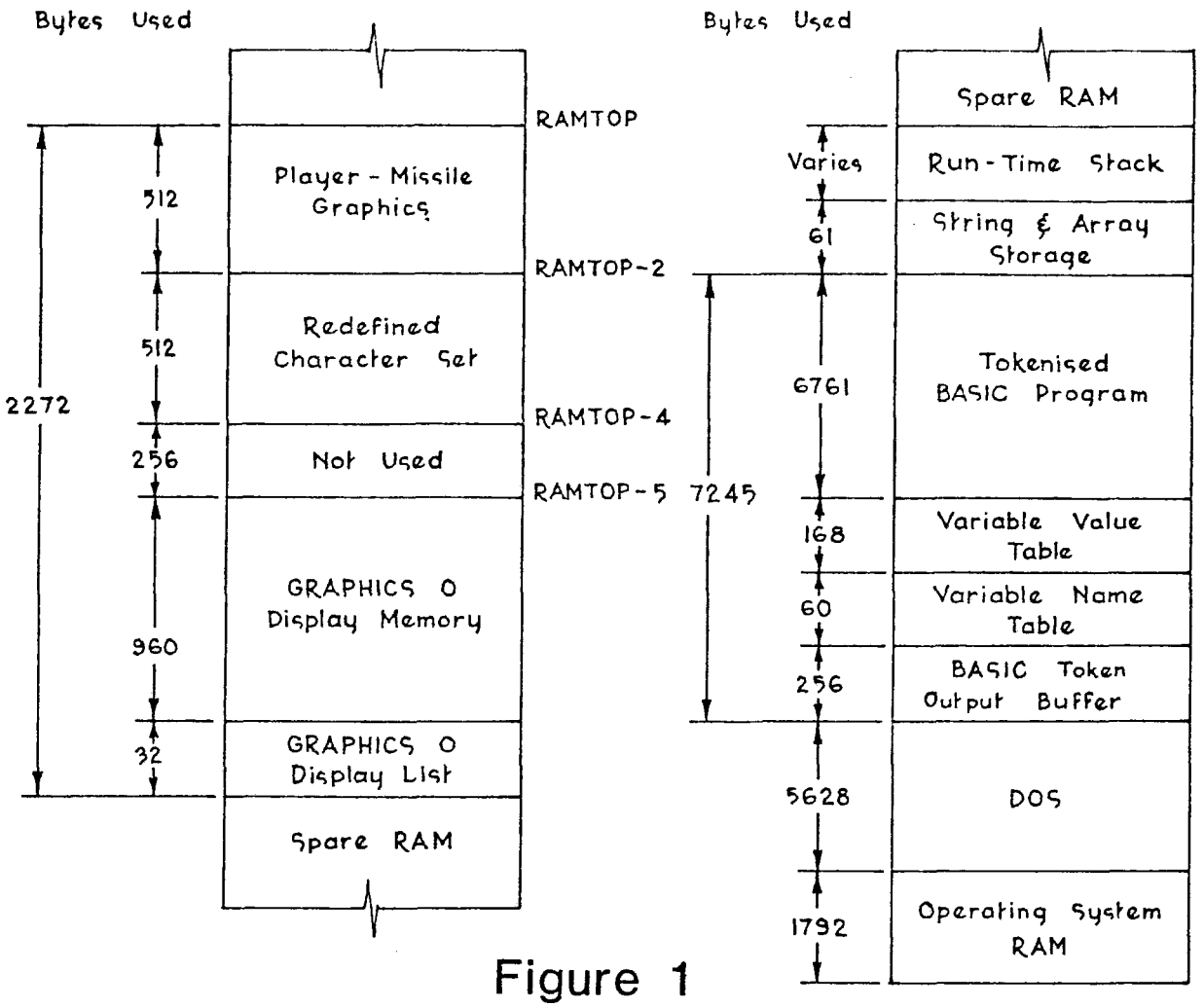


Figure 1

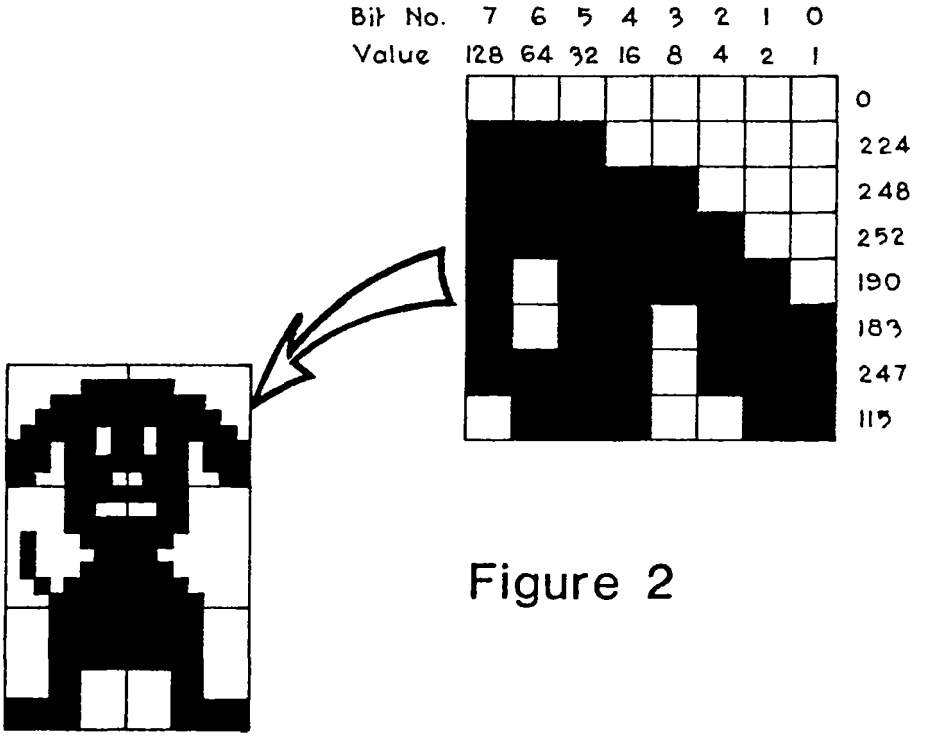


Figure 2

to clear out the same amount of memory, but these may have caused a disturbing flash of blue from the background colour of the text portion of their displays. It is care to minor details like this that makes a really professional program.

Only Player 0 is used (for the bone) and this is double width. Its shape is read directly into the Player 0 area (START+512 to START+639) at line 2140.

Vertical Blank Interrupt Processing

The vector to the Vertical Blank Interrupt (VBI) service routine is set by the call X=USR(1536) as the very last function of the initialisation code. See the assembled source code for more details. For further information on how a VBI works, see "Flashing Cursor" in the December 1982 issue of INSIDE INFO.

Once set up, the VBI reads the joystick to determine whether it has been pushed left or right and moves the bone accordingly. As this is done 50 times per second regardless of what else is happening in the program, it results in a beautiful smooth motion. Again refer to the assembled source code for more details.

Redefined Character Set

2 pages of the character set in ROM (i.e. 64 characters or 512 bytes) are copied to the new locations in our reserved area. This process is speeded up by utilising a machine language routine stored in ML$. (Refer to the assembled source code to see how it works.) It was stored in a string so that you can easily incorporate it into any of your own programs. The general form of the call is X=USR(ADR(ML$),57344,CH). 57344 is the address of the start of the character set in ROM. Change this to 57856 if you wish to copy the second half of the character set (i.e. lower case letters and control graphics characters). CH is the address of the start of your new character set. You may also use this routine to copy the whole of the character set (i.e. 128 characters or 1024 bytes) by changing the last number of the DATA statement in line 2170 from 2 to 4. This indicates how many pages are to be copied.

The Operating System pointer CHBAS at location 756 [$2F4] tells ANTIC where the character set starts. Note that we do not change this until after the instructions are cleared from the screen, otherwise some of the letters would change into parts of little doggies while we were trying to read them. Very annoying!

When the character set has been copied, we redefine 34 of the 64 characters. This is the most time consuming part of the initialisation. Allocating which characters were to be changed was quite a challenge, as the doggies required 34 characters, the score required 10 characters and the titles and various messages I wanted to print required about 24 characters - a total of 68 characters, but only 64 characters are in a GRAPHICS 2 character set. I couldn't delete any of the parts of the doggies or the digits for the score, but by carefully rewording my messages and re-allocating some of the characters for the doggies, I was able to reach an acceptable compromise. Hence messages like GREAT STUFF instead of EXCELLENT, as the X in EXCELLENT became part of a doggie. Every character except the comma is used at some time somewhere in the screen display.

When the new characters are put together in the correct pattern, they will form the shapes for the various doggies. The shape of each doggie is made up of 6 characters in a 2 by 3 grid. As an example, the stationary doggie is shown in Figure 2. The characters for each shape are stored in DATA statements. Lines 1000-1013 are for the white doggies and lines 1100-1113 are for the brown doggies. There must be at least 6 characters in each DATA statement, so the trailing blank spaces in lines 1010-1013 are replaced by inverse blanks. This is important. The program will crash and you'll be given an error message otherwise. The DATA statements in lines 1100-1113 are exactly the same as lines 1000-1013, but in inverse video. This is how we get the 2 different coloured doggies. A major subroutine of the program occurs at lines 200-220. It prints the string DOG$ at position X,Y. DOG$ is determined by the varible LINE. By changing LINE, we can change the shape of the printed doggie.

The current position of the doggies is stored in P$ (1 for a white doggie, 2 for a brown

doggie and 0 for the blank space). By comparing P$ with the final position represented in F$ (line 440), we can determine if the end has been reached.

Attract Mode

It could be said that Doggies is constantly in attract mode, as something is always happening irrespective of what the user is doing. Every time through the main loop, the program checks to see if the fire button has been pressed. If it hasn't, then it randomly selects a doggie and moves him in accordance with one of 3 randomly selected subroutines. These have the effect of making him bark, wag his tail or stomp his feet. Even when the game is over, the attract mode continues whilst waiting for you to press START for another game.

The ability to move a doggie when the game is over would cause havoc. This is averted by setting a flag called ATTRACT. When ATTRACT is 0, it indicates that a game is in progress and the GOTO in line 80 makes sure the fire button is checked. When the game is over, ATTRACT is set to 10 so that line 80 will jump past the fire button routine.

Colours

The display uses a total of 6 colours in a very pleasing combination. The black background is coloured by the Background colour register and the scungy yellow bone is coloured by the Player 0 colour register. Playfields 0 and 2 are used for the white doggies and brown doggies respectively. Playfields 1 and 3 are used for the green writing and blue writing respectively (you won't see the blue until the game is over). Green writing can therefore be achieved by printing a message in lower case and blue writing can be achieved by printing in inverse lower case. A problem arises here when we wish to print the score, as the digits 0 to 9 are in the range of characters coloured by Playfield 0, but we want them to be coloured by Playfield 3. We therefore need to convert the score to a string and manipulate the individual digits to change their colour from white to green. This is done in line 430.

The colours show up fairly well on a black and white TV and the only evidence of colour "bleeding" is from the brown doggies. This is an unavoidable problem common with dark luminances. It can only be overcome by careful selection of colour combinations.

Sounds

BARK$ contains the range of tones for the doggies' cute barking sound used in line 500. This was one point that Stan Ockers was not pleased with, but I could not come up with anything better. Apart from which, I quite like it.

The subroutine for the footsteps sound in lines 900-910 is also unusual. The odd numbered distortion value causes the speaker to click, then silence. Turning the voice off then causes another click. When executed together, the 2 clicks are indistinguishable and sound as though they are combined to form one single louder click. Prior to entering this subroutine, the variables V and INC are specified. V is the volume of the footsteps and INC is the increment by which the volume is increased or decreased. In this way the footsteps may stay at a constant volume while the doggie is stationary (INC=0), decrease as the doggie walks away (INC=-1) or increase as the doggie approaches (INC=1). Using a common subroutine ensured that timing of the footsteps is always the same. This subroutine is also the reason for the apparent duplication of DATA statements at lines 1010-1013 and 1110-1113.

Scoring

Your score is incremented every time a doggie is moved. The object is to move all the doggies in as few moves as possible. 15 moves is the best you can do it in, but scores of around 21 are more common.

There is a sort of bug in the scoring routine. Even though it will never be encountered under normal use, I mention it here as a good example of deciding where to "draw the line" for certain error conditions. The score's colour conversion mentioned above will crash if there are more than 3 digits in the score. I decided on a maximum of 3 digits for 2 reasons. Firstly, more than 3 digits would wraparound onto the next line and mess up the display. Secondly, you'd have to be a complete moron to need 1000 moves. Even the smart alec who intentionally tries to crash the program would have to move one doggie every 5 seconds for

an hour and a half before the program would complain. If the idea of this bug still worries you, just increase the size of MOVE$ in line 2070 to 4. Our hypothetical smart alec would then be moving doggies for 14 hours!

Human Engineering

One of the most important aspects of any program is its human engineering. Here is where Doggies excells. It couldn't possibly be any simpler to use! After initialisation, you need only press the START key for the game to begin. In fact, you can then press START again at any time and the game will restart. This comes in handy if you can see that you've made a mistake and don't want to carry the game through to completion knowing that you'll get a bad score.

The bone is controlled by pushing left or right on a joystick plugged into Port 1. If it goes off the screen, it will wrap around to the other side. A doggie is selected by placing the bone under the doggie you wish to move and pressing the fire button. The bone is not very fussy when it comes to selecting a doggie. It doesn't have to be directly under a doggie, just make sure that at least half of it is under the doggie you wish to move. If it is exactly in the middle of 2 doggies, a choice still has to be made, so it opts for the doggie on the right.

The program won't let you make an illegal move. I won't spoil the surprise by telling you what happens, so try it and see for yourself.

As there is no keyboard input, the well known random colour switching would normally come into effect after just under 11 minutes. As Doggies is such a compulsive game, you will quite likely be playing for well over 11 minutes, so the random colour switching must be catered for. Any legal input at all (i.e. pressing START, pushing the joystick left or right or pressing the fire button) will reset the ATRACT flag at location 77 [$4D], thereby avoiding the random colour switching. If however you have to answer the 'phone or you go on holidays and forget to turn the computer off, the random colour switching will still be enabled as usual. This will protect the phosphors in your TV set. When you come back, just move the joystick, press the fire button or press START and everything will return to normal.

All in all, the human engineering is so well done that even the youngest child (or the oldest computer critic) can enjoy playing Doggies.

Conclusion

Well, that about wraps it up for Doggies. If you've read this far and haven't keyed it in yet, then you're missing out on quite a treat. Doggies is undoubtedly the most professional game we've printed so far, so give it a go. We'll publish the solution next issue.

* * *

FOOTNOTE

Stan Ockers' Chicken was published in the Eugene A.C.E. Newsletter in February 1982 and reprinted in ANTIC Vol.1 No.1 April 1982. As a lot of people typed the program in from ANTIC, you may be interested in the following. Someone mentioned that there is a bug in the program whereby if you plug a joystick into Port 2, you can actually move the cars and cause all sorts of havoc! Now if you subscribed to the excellent A.C.E. Newsletter, then you'd have known the fix for this problem before the ANTIC version was even printed! The following is from the March 1982 issue of the A.C.E. Newsletter:

> "After going to all that trouble to set up the masks for joysticks in Chicken, I forget to mask them out. It's a simple matter to fix. The sequence 12,0,0,0 in line 52 should be 12,15,15,15.
>
> Someone asked why the chicken can't go sideways. Easily enough done. Change the 12 in the same line to a zero. A warning though! The chicken can go to the edge limits and score like mad because the cars don't go that far. You'll eventually rack up so many points that you'll get an error message as the asterisks try to print off screen.
>
> - Stan Ockers"

```
                0100 ;################################
                0110 ;#   SOURCE CODE FOR MACHINE    #
                0120 ;#  LANGUAGE STUFF IN DOGGIES    #
                0130 ;#      by Garry Francis        #
                0140 ;# Published by Atari Computer  #
                0150 ;#    Enthusiasts (N.S.W.)      #
                0160 ;#        February 1983         #
                0170 ;################################
                0180 ;
                0190 ;ROUTINE TO SET VBI VECTOR
                0200 ;
E45C            0210 SETVBV =    $E45C
0000            0220         *=  $0600      ;Origin
0600 68         0230         PLA            ;Pop no. of arguments off stack
0601 A00A       0240         LDY  #START&$FF ;LSB of VBI service routine
0603 A206       0250         LDX  #START/256 ;MSB of VBI service routine
0605 A907       0260         LDA  #$07      ;Indicates deferred VBI
0607 4C5CE4     0270         JMP  SETVBV    ;Set vector
                0280 ;
                0290 ;VERTICAL BLANK INTERRUPT SERVICE ROUTINE
                0300 ;
D300            0310 PORTA  =    $D300
004D            0320 ATRACT =    $4D
D000            0330 HPOSP0 =    $D000
E462            0340 XITVBV =    $E462
00D1            0350 BONE   =    $D1
060A AD00D3     0360 START  LDA  PORTA      ;Read joystick
060D 48         0370        PHA            ;Save on stack
060E 2908       0380        AND  #$08      ;Joystick pushed right?
0610 D006       0390        BNE  LABEL1
0612 E6D1       0400        INC  BONE      ;Yes...increment position of bone
0614 68         0410        PLA
0615 18         0420        CLC
0616 9007       0430        BCC  LABEL2    ;Forced branch
0618 68         0440 LABEL1 PLA            ;Restore joystick reading
0619 2904       0450        AND  #$04      ;Joystick pushed left?
061B D00B       0460        BNE  FINISH
061D C6D1       0470        DEC  BONE      ;Yes...decrement position of bone
061F A900       0480 LABEL2 LDA  #$00
0621 854D       0490        STA  ATRACT    ;User alive, so reset attract flag
0623 A5D1       0500        LDA  BONE      ;Get position of bone
0625 8D00D0     0510        STA  HPOSP0    ;Change it in hardware register
0628 4C62E4     0520 FINISH JMP  XITVBV    ;Exit the VBI
                0530 ;
                0540 ;SUBROUTINE TO TRANSFER 2 PAGES OF CHARACTER
                0550 ;SET FROM ROM TO USER DEFINED AREA.
                0560 ;
                0570 ;Code is fully relocatable, so may be stored
                0580 ;in a string (ML$ in Doggies). Assembled to
                0590 ;this region for convenience only so that it
                0600 ;it could be printed in the magazine.
                0610 ;
                0620 ;Call from BASIC using:
                0630 ;    X=USR(ADR(ML$),57344,CH)
                0640 ;where 57344 is starting address of character
                0650 ;set in ROM and CH is starting address of new
                0660 ;character set.
                0670 ;
062B 68         0680        PLA            ;Pop no. of arguments off stack
062C 68         0690        PLA
```

```
062D 85CC   0700        STA  $CC       ;MSB of source address
062F 68     0710        PLA
0630 85CB   0720        STA  $CB       ;LSB of source address
0632 68     0730        PLA
0633 85CE   0740        STA  $CE       ;MSB of destination address
0635 68     0750        PLA
0636 85CD   0760        STA  $CD       ;LSB of destination address
0638 A202   0770        LDX  #$02      ;No. of pages to move
063A A000   0780 LABEL3 LDY  #$00      ;Zero the counter
063C B1CB   0790 LABEL4 LDA  ($CB),Y   ;Get a byte
063E 91CD   0800        STA  ($CD),Y   ;Move it
0640 88     0810        DEY            ;Decrement counter
0641 D0F9   0820        BNE  LABEL4    ;Finished a page?
0643 E6CC   0830        INC  $CC       ;Yes...increment MSB of source
0645 E6CE   0840        INC  $CE       ;...and MSB of destination
0647 CA     0850        DEX            ;Decrement no. of pages
0648 D0F0   0860        BNE  LABEL3    ;Finished yet?
064A 60     0870        RTS            ;Yes...return to BASIC
064B        0880        .END
```

PROGRAM LISTER

by Peter Bamford

Use this short utility to control the rate that your program lists to the screen. After typing it in, LIST it to disc or cassette. It can then be used at any time by ENTERing it into the program you wish to list. Type GOTO 32000 and the screen will fill with the first "page" of your program. You can now list your program at your leisure by pressing SELECT to go forward one page or OPTION to go backward one page. You can also press START to (what else?) return to the first page. To edit your program, press the BREAK key, then edit lines in the usual fashion. Type GOTO 32000 again to re-list the program. You can try CONT, but it usually won't work properly because your editing will have reallocated the location of your lines within memory.

If you want each line separated to make it easier to read, remove the "? CHR$(28)" from line 32030 (this is the up arrow). It is also a simple matter to alter the program to list continuously with a delay between each line by adding a delay loop in line 32030 and deleting all the unnecessary steps.

The program is stored from lines 32000 onwards to conform with our listing conventions, but I personally find the program easier to use by changing the line numbers and GOTO's to the range 1-9. The program was purposely restricted to 9 lines for this reason. The lister then works by typing RUN, but your main program can only be executed by typing GOTO 10 or GOTO 100 or whatever.

The utility works by reading the tokenised form of the program which starts at the address given by locations 136 (low byte) and 137 (high byte). The first 2 bytes of each line give the line number and the third byte is the offset to the start of the next line.

```
1 REM ##############################
2 REM #      PROGRAM LISTER        #
3 REM #      by Peter Bamford      #
4 REM # Published by Atari Computer #
5 REM #   Enthusiasts (N.S.W.)     #
6 REM #       February 1983        #
7 REM ##############################
32000 GRAPHICS 0:CLR :DIM A(300):TRAP
32080
32010 I=0:X=PEEK(136)+256*PEEK(137)
32020 ? CHR$(125);:A(I)=X:FINISH=0
32030 LIST PEEK(X)+256*PEEK(X+1):X=X+P
EEK(X+2):IF PEEK(84)<20 THEN ? CHR$(28
);:GOTO 32030
32040 IF PEEK(53279)=3 AND I THEN I=I-
1:X=A(I):GOTO 32020
32050 IF PEEK(53279)=5 AND  NOT FINISH
 THEN I=I+1:GOTO 32020
32060 IF PEEK(53279)=6 AND I THEN 3201
0
32070 GOTO 32040
32080 TRAP 32080:FINISH=1:? :? "END OF
 LISTING";CHR$(253):GOTO 32040
```

Odds & Sods

A regular column by Garry Francis

The 1st Australian Personal Computer Show will be held at Centrepoint in Sydney from 10-12th March 1983. The hours are 9 A.M. - 7 P.M. on Thursday and Friday and 9 A.M. - 5 P.M. on Saturday. Access is via the lifts at the Market Street entrance. The normal admission price is $3, but if all goes well, you should have a half price ticket enclosed with this magazine enabling you to get in for $1.50! A.C.E.(N.S.W.) will be at the Computer Club Corner, so come along and say "Hello" and give us some encouragement, especially all you out-of-towners. See you there!

* * *

The blank discs mentioned in the last issue were an outstanding success, so we've ordered some more. Just as a reminder, these are available for $28 per box of 10. Unfortunately though, we got our fingers burnt with the postage costs, as this varies widely depending on the destination. Please allow sufficient for return postage. As a guide, this will be about $1 within N.S.W. and $2.50 elsewhere. If you live overseas, the exhorbitant postage doesn't make it a worthwhile proposition.

* * *

Cassette users needn't feel left out. After a bit of bargaining, we can now offer members a box of 10 C45 audio cassettes for only $9! They come with their own self adhesive labels. Postage is roughly the same as for discs.

* * *

Futuretronics has just announced that the ATARI 800 is to be sold with 48k installed as standard at no extra cost and a 48k upgrade is available for 400's at (I think) $120. Therefore we will no longer be offering the 48k board to members.

* * *

Datasoft Inc. have made an intelligent move in their search for new ATARI products by contacting users' groups directly. In a recent letter, they offered to contribute $500 to $2000 to our users' group fund if we referred a software or hardware product or an assembly language programmer to them. The letter said, in part:

> "We will pay all programmers we hire competetive salaries plus a bonus based on how well the product sells, and relocation costs if applicable. They might also have the opportunity to program well known products like SEGA's (R) ZAXXON, the #1 arcade game for 1982, MOON SHUTTLE by Nichibutsu, and other programs based around cartoon characters and TV personalities."

Sounds interesting doesn't it? Contact the secretary if you think you fit the bill. And if you do think you're that good, then how about sparing a little time to write a series on assembly language programming for the benefit of other members?

* * *

G. Wragg is offering a stand alone, Centronics compatible 64k printer buffer with reprint capability for $278 and a 32k Epson printer buffer for $188. Both prices include post and packing. For further details, write to:

G.L. Wragg
4 Pryton Court,
Balwyn,
Vic. 3103

Hints & Tips

If you buy good quality single sided, single density discs for your disc drive, then there is no reason in the world why you shouldn't use the flip side as well. Simply use a paper punch to add another write protect notch on the opposite edge to the one that's already there. It is 33mm from the corner to the centre of the hole. Turn the disc over and format it the same as usual. I have done this to my entire disc collection to effectively double it and haven't yet had a disc that couldn't be used on both sides.

* * *

Do not use head cleaning discs on an ATARI disc drive, as these generally use an abrasive to remove any accumulated dirt and may therefore damage the head. The following advice comes direct from ATARI in "The ATARI Connection" Vol.1 No.1:

> "The heads of your ATARI disc drive can be damaged by some head cleaning diskette kits. We advise owners of these disc drives not to use any head cleaning diskettes. Clean the heads much as you would clean the head of a tape deck. Gently wipe the head with a bit of cotton soaked in denatured alcohol. Let it dry for 30 minutes before using the drive."

* * *

I used the 410 Program Recorder constantly for 6 months before getting a disc drive and in all that time, I never had a single problem! Honest! Not even a bad load! A recent article submission from a member had 6 versions of a program on a cassette - every one of which was bad. I then realised that many people are not aware of good cassette usage techniques, so here are Garry's rules for much improved cassette performance.

- The ATARI outputs an audio signal, so make sure you use high quality audio cassettes - the cheapies you buy at the local supermarket are no good. Don't use "digital" or "computer" cassettes.
- Use the shortest cassettes you can get (C30 or C45). C60 should be considered a maximum as the longer cassettes use thinner backing material and are therefore more prone to stretching.
- Don't use chromium dioxide or other "metal" based cassettes as they are very abrasive and the 410's recording heads are not built to take it.
- Before using a new cassette, fast forward it for its full length (both sides) to ensure that it is not stuck together at any point, hence causing a "drop out".
- Ensure a cassette is fully rewound before saving a new program on it. The program will then be easier to find. Avoid tapes with exceptionally long leaders for this reason.
- Do not leave gaps between programs!
- The maximum number of times you can safely use a cassette is about 50 reads. After this period, the strength of the signal on the cassette begins to fade due to "thin spots" as the magnetic coating wears away. Save your program on a new cassette if you still wish to use it after this period.
- Keep your cassettes away from any magnetic forces such as TV sets, radios, speakers, etc. Also keep them protected from prying fingers, dust, extremes of temperature and other environmental hazards.

* * *

This one's from Gregory Marthick...
If you type a lot of programs from books and magazines, but do little or no programming yourself, then you are possibly not aware of all the abbreviations that can be used in ATARI BASIC. Admittedly, you probably know they exist, but don't always use them because either you cannot remember them or you just never learnt them in the first place. Well I got fed up with this, so I made a list of all the abbreviations (in alphabetical order) and taped it on the wall next to my ATARI. A full list of abbreviations is given in Appendix A of the BASIC Reference Manual. Admittedly, some of them seem pretty silly (what's the difference between typing CLOA. and CLOAD?), but some of them are handy too. [There is actually good reason for some of those seemingly silly abbreviations. See "Insight:ATARI" in the January 1983 issue of COMPUTE! - GF]

While I'm at it, I would like to thank Garry for his tip on using the Assembler Editor cartridge as a text editor way back in Issue No.1. I recently had an old printer given to me, so by buying a Macrotronics parallel interface and using the Assembler Editor, I've got a good printing setup for a total outlay of only $99.

Questions & Answers

Q: How can I get rid of the question mark prompt issued by the INPUT statement?

- Gregg Ramsey

A: You can't. The question mark prompt is a built in feature of nearly all languages (BASIC, Fortran, etc.) that allow interactive input. It is their way of letting the user know that the computer is waiting for a response. As ATARI BASIC is in ROM, it can't be changed. However, there are several techniques that come to mind that will allow you to overcome this.

- Express your input as a question and the question mark prompt won't look out of place, e.g. use "WHAT IS YOUR NAME?" rather than "ENTER YOUR NAME?".
- Use a redefined character set and redefine the question mark so that it appears as either a blank space or as a different prompt. If you need to use the question mark somewhere else in the program, then redefine another unused character (such as ^ or @) to look like it.
- Use GET instead of INPUT. This is a great technique for writing idiot proof input routines, but give a lot of thought to what you're doing. Cursor control keys in particular will need special handling. Try the following example. It only allows you to use upper case alphabetic characters, BACK SPACE to correct mistakes and RETURN to end the input. BACK SPACE and RETURN can't be used unless at least one character has been entered and you cannot enter more characters than the dimensioned length of the string. Even though the routine seems almost foolproof, the TRAP is still necessary in case of the unlikely event that some smart alec presses CONTROL 3.

```
10 GRAPHICS 0:OPEN #1,4,0,"K:":DIM A$(20)
20 TRAP 20:L=0:A$="":? :? "ENTER YOUR NAME...";
30 GET #1,A:IF A>64 AND A<91 AND L<20 THEN ? CHR$(A);:L=L+1:A$(L)=CHR$(A):GOTO 30
40 IF  NOT L THEN 30
50 IF A=126 THEN ? CHR$(A);:A$(L)="":L=L-1:GOTO 30
60 IF A<>155 THEN 30
70 CLOSE #1:TRAP 40000:? :? "YOUR NAME IS ";A$:END
```

- I suspect you may also be able to do your own input independant of BASIC by setting up the necessary parameters in the relevant Input/Output Control Block (IOCB) and calling Central Input/Output (CIO) directly, but I haven't had time to play around with this.

* * *

Q: I'm looking to buy a good printer, not too expensive, but good quality, an Epson maybe? I was thinking about buying it from U.S.A. (mail order). Can you recommend a place that will give me a good deal?

- Ernie Sugrue

A: No. Can anyone else help?

* * *

Q: I'm having trouble with a program from the August 1982 issue of COMPUTE! called "System Clock for the ATARI". It works fine except that it is set for the 60Hz frequency. How can I correct it for our 50Hz frequency?

- Phil Strong

A: The program in question updates its clock display during the vertical blank interrupt. If you disassemble the VBI routine, you should be able to work out where the discrepency lies and fix it by adjusting only 1 or 2 bytes. An Editor's job is a hectic one, so don't expect me to do it for you. I would also suggest that you send your query to COMPUTE! in order to make their readers aware of the problems that they are creating for we poor Aussies by not allowing for PAL systems in their programs. If anybody else has problems with programs in commercial magazines, send your query to the magazine concerned, not to us! This is the first and last time that I shall print a question relating to a commercial publication.

* * *

Q: I know you can carry out a lot of the options on the disc menu directly from BASIC using XIO commands (viz. format, lock, unlock, rename and delete), but how can I get a disc directory from BASIC?

- Gregg Ramsey

A: Whenever a disc file is opened for access, the 1st auxilliary byte (aexp1 in the OPEN statement) specifies what mode you wish to use, e.g. 4 for read, 8 for write and 12 for append (i.e. read and write). The disc directory may be opened just like any other file by specifying this byte to be 6 (see pages 26-27 of the BASIC Reference Manual). The filespec will be whatever file(s) you wish to look for in the directory. You may use wild cards, so that to get the whole directory, use a filespec of "D1:*.*". You can now read each entry from the directory using an INPUT statement. The following bare bones program illustrates these points by printing the whole directory to the screen without any fancy formatting. Try it!

```
10 GRAPHICS 0:DIM A$(17):OPEN #1,6,0,"D1:*.*":TRAP 30
20 INPUT #1,A$:PRINT A$:GOTO 20
30 CLOSE #1:END
```

* * *

Q: I have Space Invaders on cassette (approx. 12 months old) which has failed to load. It's an autoboot cassette where you have to hold the START key down while turning the computer on. It always comes up with "BOOT ERROR". Can anyone advise as to how I might save the program from the scrap heap?

- John Trigge

A: If the cassette will no longer load, then it's probably due to a bad record somewhere. This may have been brought about by excessive use or the cassette not being looked after properly. You can determine which record is bad by counting off the number of "bursts" of data until the BOOT ERROR message comes up. It is then simply a matter of writing a little routine to write the whole program into memory using (say) burst input/output, disassembling the area that is bad and trying to fix it up. If you know enough about the ATARI to know what I'm talking about, then you probably wouldn't have asked the question in the first place, but now that you know what's involved, you'll probably see that it's just not worth the effort. Does anyone have any better suggestions?

Software Exchange

The Software Exchange is now formally in operation. Following extensive discussion at the last committee meeting, it was decided to run the Exchange along the lines set out below. These were formulated to provide the maximum benefit at the minimum price. We are supporting only high quality programs rather than the usual rubbish that users' groups are famous for. We are also encouraging original program submissions by offering even more attractive benefits to authors. Keep in mind that these are only guidelines and may change in future if experience shows that we can improve the benefits to members.

Submissions: Programs must be submitted on disc or cassette (disc is preferred) and must be accompanied by a completed copy of the Software Submission Form at the end of this issue. If a Software Submission Form does not accompany the program or the declaration is not signed, then the program will be rejected. This is our protection against copyright infringement. (Please note that programs in magazines are not public domain unless it specifically says so! At this time, ANTIC is the only magazine with public domain programs.) All submissions should be addressed to:

Software Exchange
Atari Computer Enthusiasts (N.S.W.)
G.P.O. Box 4514,
Sydney,
N.S.W. 2001

Alternatively, they may be handed direct to John Massara or Gregg Ramsey at any of the meetings.

The programs will be reviewed and the original media will be returned to the author. If the program is full of bugs, not considered of interest to members or of generally poor quality, it will be rejected outright. If it has potential, but needs improvement or polishing up in some areas, the author will be advised. He may then carry out any changes at his discretion and resubmit the program. The second or subsequent submissions won't need a new Software Submission Form unless there are substantial changes.

When a program is accepted, the author will be given a "Credit Note" (this applies only to original programs). This entitles him to any selection from the Software Exchange at half its normal price. The same deal applies to any program published in INSIDE INFO and subsequently put in the Software Exchange.

If time permits, I will produce an Author's Guide with enhanced details in the near future. In the meantime, please keep the following 2 points in mind:

1. The Software Exchange will be disc based, but we intend that all programs will be available on cassette unless this is absolutely impossible for the particular application. Therefore, please write all programs so that they are compatible with both disc and cassette based systems.
2. Line numbers 0-9 are reserved for title, author, etc. Lines 10-29999 should be used for any "normal" program. Lines 32000-32767 should be used for any utility which is expected to co-reside with a normal program (such as Program Lister in this issue). This is a defacto standard which we have already adopted in INSIDE INFO and is generally adhered to throughout the ATARI community. The reason that lines 30000-31999 are not used for normal programs is that a couple of commercial utilities use this region. It is therefore better to avoid these lines altogether and ensure full compatability.

Buying from the Software Exchange: If all goes well, the first releases will be announced in the next issue of INSIDE INFO. Do not pester us until then. When released, the prices will be $10 for a disc (or $5 with a Credit Note) and $8 for a cassette (or $4 with a Credit Note). Postage is extra.

The programs will usually be arranged so that a number of programs of a similar category (games, utilities, educational, etc.) are on the same disc or cassette. The number of programs will vary from one to about ten, depending on their size. There may also be

"special" discs and cassettes such as Best of INSIDE INFO, Best of Eugene A.C.E., ANTIC's public domain discs and so on.

Well, that's about it for now. The sooner you send programs in, the sooner we'll have stuff to sell, so hop to it!

- Garry Francis

New Members

Gordon Anderson, Newport
Jamie Athas, Maroubra
Arthur Banks, Tyntynder South, Victoria
Darryl Barlow, Sylvania Waters
Peter Barrett, Wahroonga
Jeffrey Benning, Page, A.C.T.
Robert Brand, Woollahra
Terrence Burgess, Mount Ousley
Michael Chan, Ermington
M. Coper, Kensington
Ian Crane, Rockdale
L. Csenderits, Turramurra
Timothy Cumpston, Coogee
D. Denshire, Belmore
Mark Friezer, Waverley
Chris Garrett, Coffs Harbour
Colin Grace, Vaucluse
Keith Hall, Cremorne
Greg Hamilton, Highett, Victoria
David Kingston, Avalon Beach
Wendy Littlejohns, Macquarie Fields
John Macdonald, Kingsgrove
Matthew Maguire, Belmont North
J. Mastenbroek, Roma, Queensland
John Mattes, Seaforth
Robert Michaelis, Terrey Hills
Daniel Nesci, Glebe
Ingo Oestreich, Earlwood
Donald Pereira, Ryde
Nye Perram, Greenwich
C. Porter, Ashfield
Ken Radcliffe, Point Clare
Michael Rae, Wallsend
Karl Rasmusson, Toowoomba, Queensland
Roberto Romano, Bellevue Hill
Jeff Rushton, Ingleburn
Peter Sheedy, Killarney Heights
Allan Smyth, Hurstville
Glen Spruce, Condobolin
Richard Stephenson, Douglas Park
Doug Stone, St. Ives
Chris Watts, East Malvern, Victoria
Anthony Williams, Bardwell Park
Richard Wilson, Paddington
Bruce Wolfe, New Plymouth, New Zealand

REPORT FROM U.S.A.

by Brenton Vettoretti

As many readers of INSIDE INFO will know, I have just returned from a 3 month stay in the Sacramento area of U.S.A. Before I went over, I promised Garry a report on my findings, but to be truthful, I was very disappointed with the personal computer scene over there. I would have preferred to let this article die a slow and natural death, but our persistent Editor had other ideas. [Well you promised it to me and I in turn promised it to our readers, so what do you expect? - GF]

Firstly, let me point out that I was on a business trip and did not have any time for investigative snooping (unfortunately). I searched out all the computer stores in the Sacramento area, which almost sent my colleagues nuts. My first visit was to Computerland, who were not supporting the ATARI, however, the salesman directed me to ON-LINE Computer Centre - the city's biggest ATARI dealer. I dealt with a very professional rep named George Chung, who had no objections to my playing the newly released games from Sirius Software. These are now available in Australia at a reasonable price through Imagineering, so I won't review them here.

George also gave me information on the local user group known as ATARI Computer Club Encompassing Sacramento or ACCES. They have over 300 members varying in age from 8 to 80 and of both sexes. The meeting was held in the furniture department of a large department store (like Grace Bros), so the surroundings were quite novel. The lounge suites and beds were put to good use. From the plush velvet lounge, I was astonished to see 3 large projection screen TV's connected to an ATARI 800! The comfort of the seating and the ability to see the screen while sitting down were great, but that was the only thing that impressed me. The President of ACCES got up to do his bit including the dreaded piracy speech and all about the group's financial statement, like how many stamps had been bought. This went for over an hour. Boring! Things then livened up with a well presented demo of the Alien Group's Voice Box, but then the President stood up again to tell everyone that it was available in the computer department downstairs for $XXX. There were a few more demos of software with a note from the President at the end of each one to tell everyone that (you guessed it) it could be bought downstairs. At the end of the meeting, the President reminded everyone to stop at the computer department on the way out. I later found out that he was the Computer Department Manager. I am sure that not all groups suffer from the same problem, but the people of Sacramento are missing a great deal.

Whilst in the U.S.A., I was also warned about buying products through the mail order houses which advertise unbelievably low prices. There were several reasons for this. Firstly, you may never hear from them after they've received your cheque. Secondly, your order can take a long, long time to arrive as they often don't have the stuff you want in stock. And thirdly, if they don't have what you requested, they will usually send you something else a little lower in price along with a credit note to ensure that you use them again. It is unfortunate, but it does happen all too often over there. I recommend ringing the store concerned (if possible) before sending any money.

[Some of Brenton's observations are quite interesting, but as he was only in a very localised area, don't get the impression that they are necessarily typical. The point about mail order houses is a good one. I have heard about this before, so be very careful in choosing a mail order house. Ask others about their experiences and start with a small order. Don't send any large amounts of money until you are satisfied that they give prompt and honest service. Also don't order any new products until you have actually seen them reviewed in magazines, as another of their ploys is to advertise tempting software long before it is available. - GF]

Atari Computer Enthusiasts (N.S.W.)
Software Exchange
PROGRAM SUBMISSION FORM

Program name: _____

Author(s): _____

Category (Game, Utility, Education, etc.): _____

Disc filename (if applicable): _____

Language: _____

Minimum memory required (disc): _____ (cassette): _____

Peripherals or accessories required: _____

Brief description: _____

_____

_____

Loading instructions: _____

_____

_____

Instructions for use (attach a separate sheet if insufficient room): _____

_____

_____

_____

_____

May users contact you if they need help with your program? _____

I warrant that the program submitted by me and described above is:

(a) my own original work and does not infringe upon any other party's copyright.

(b) not my original work, but is public domain and may be freely used without acquiring the original author or publisher's permission and evidence of this is attached.

(c) not my original work, but the original author or publisher's permission to use the program has been obtained in writing and a copy of that approval is attached.

Signature: _____

Date: _____

Name: _____

Address: _____

_____

Phone: _____

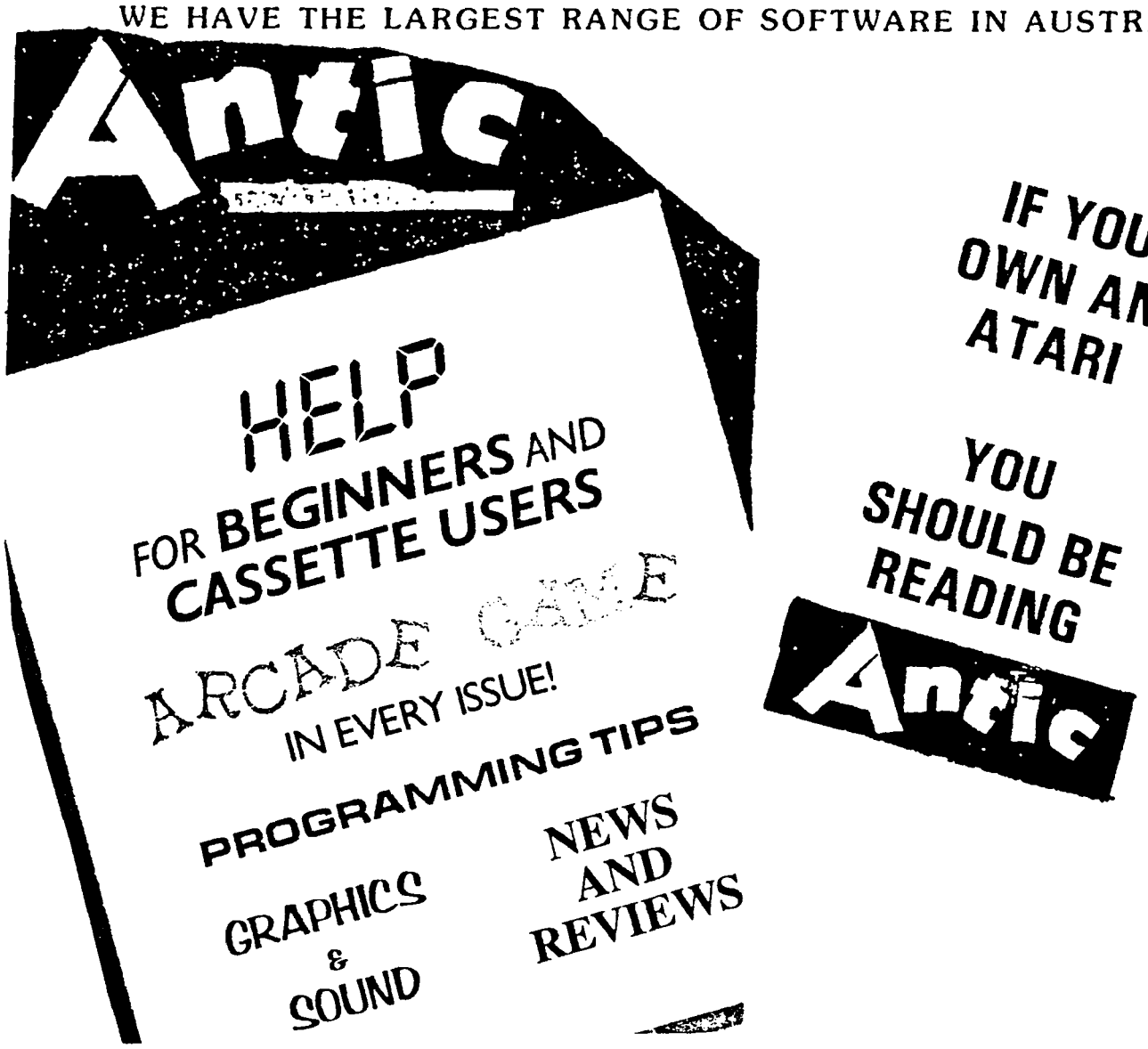
COMPUTER WARE

PTY LIMITED

All Correspondence to:
Box 268
GPO Sydney
N.S.W. 2001

Lower Ground Floor
Myer Sydney Store
George and Market Streets
Sydney
Tel: 238 9984

WE HAVE THE LARGEST RANGE OF SOFTWARE IN AUSTRALIA



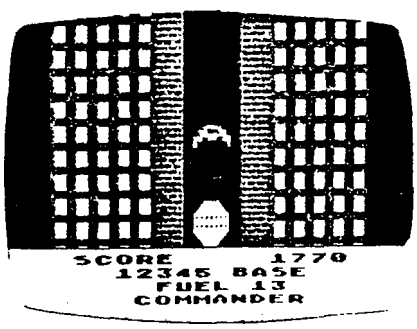
IF YOU OWN AN ATARI

YOU SHOULD BE READING Antic

WE NOW HAVE CASSETTE VERSIONS OF

Chris Crawford

CAVERNS OF MARS™



EASTERN FRONT (1941)

A one-player simulation of the German invasion of Russia (teens and up)

Cassette: 16K (APX-10050) Diskette: 32K (APX-20050)

User-Written Software for ATARI Home Computers